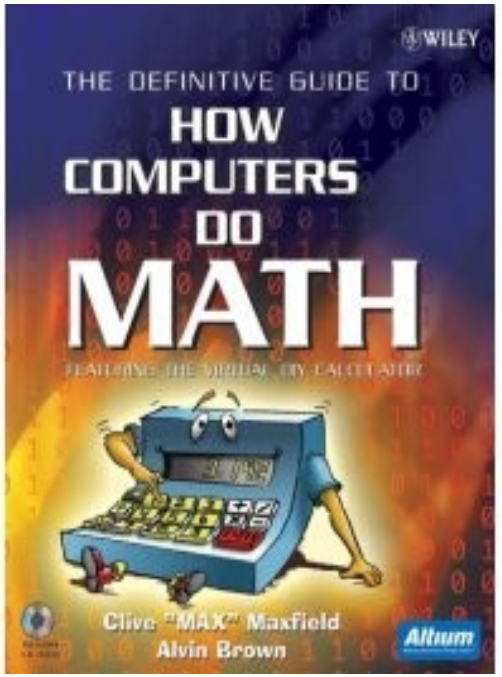
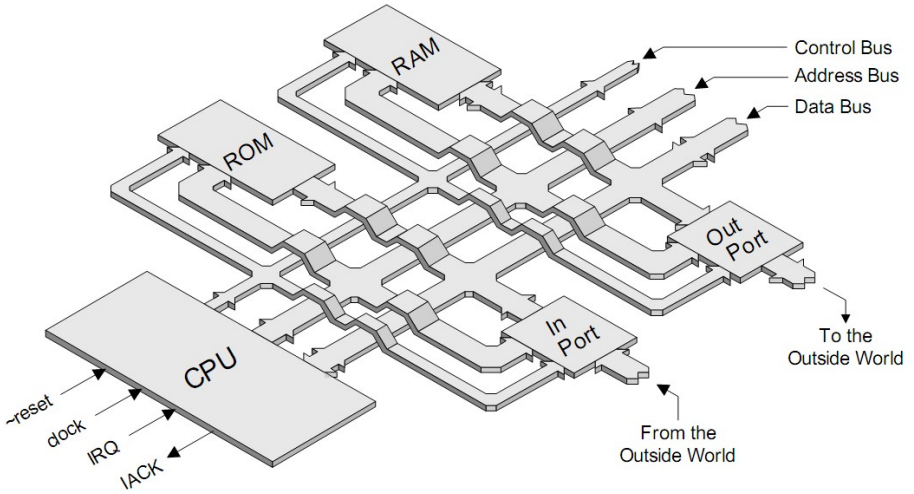
microcomputer “internals”

based on
The Definitive Guide to How Computers Do Math: Featuring the Virtual DIY Calculator
by Clive Maxfield and Alvin Brown

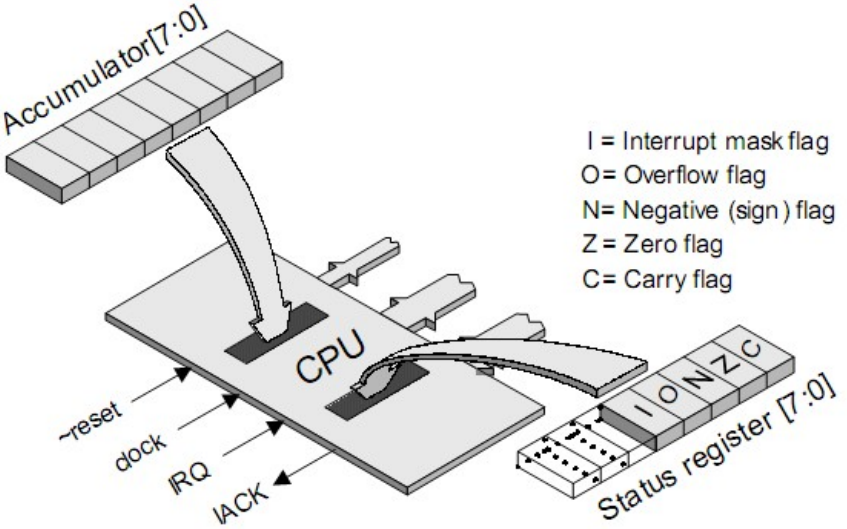


main components of microcomputer



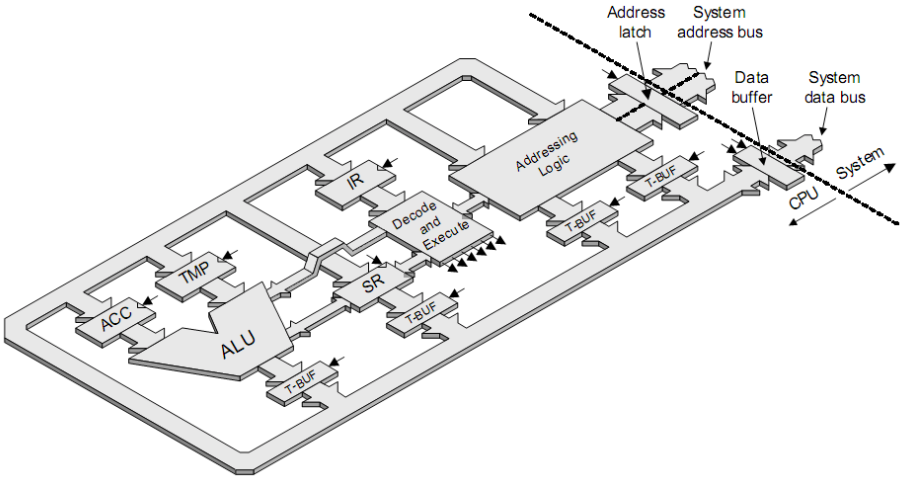
accumulator (acc) and status register (sr)

- acc
  - “in nomen est omen”
  - Gathers, piles up (intermediate) results
- sr (collection of flags)
  - N: negative/sign flag
  - Z: zero
  - C: carry
  - O: overflow
  - I: interrupt



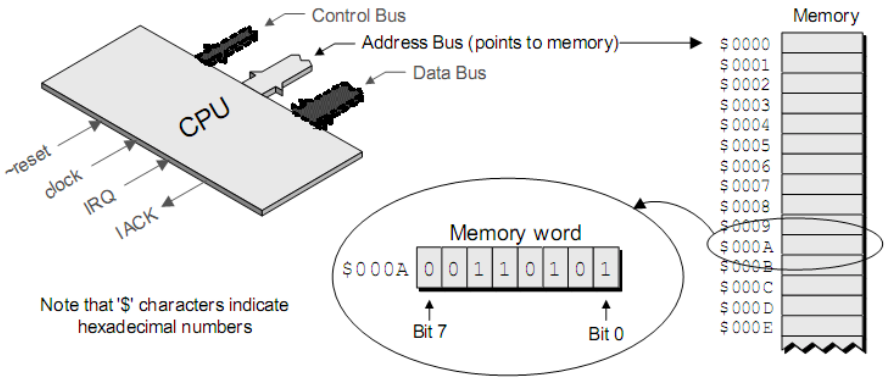
internal view of cpu

- ACC: accumulator
- TMP: temporary register
- IR: instruction register
- SR: status register



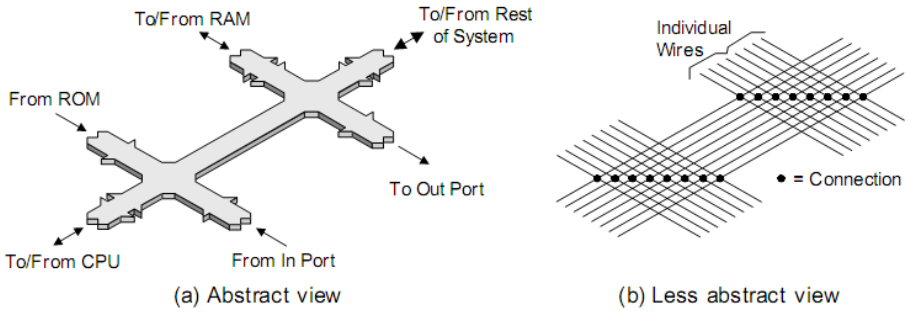
cpu with address bus

- address space
- memory width
  - data bus width



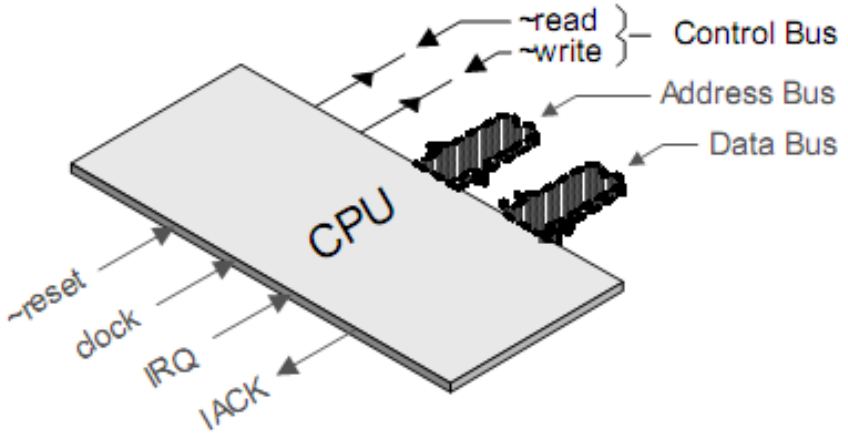
data bus

- arrows show the directions in which signals can travel



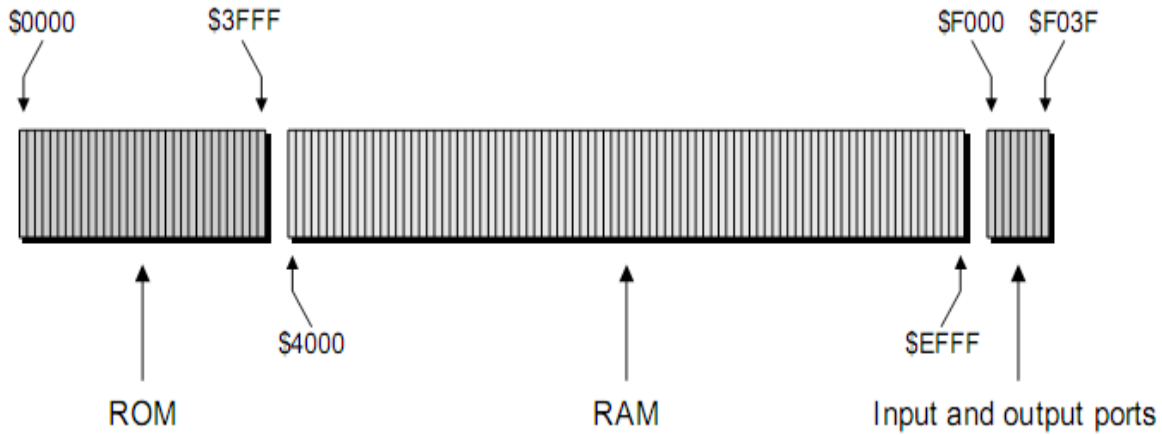
control bus

- ~read/~write
- clock
- ~reset
- IRQ: interrupt request
- IACK: interrupt acknowledge



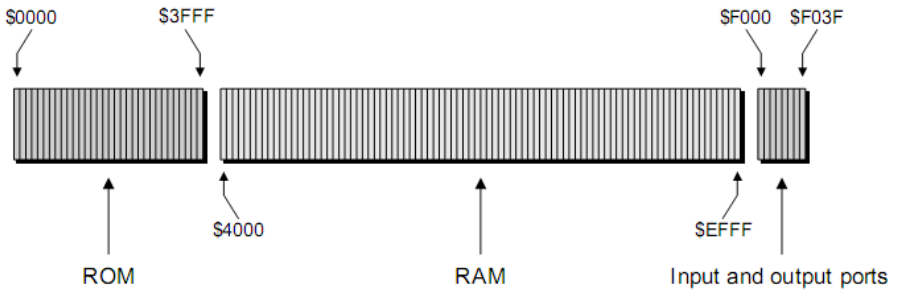
memory map/allocation

Addresses	Function
$0000 to $3FFF	ROM
$4000 to $EFFF	RAM
$F000 to $F01F	Input ports
$F020 to $F03F	Output ports
$F040 to $FFFF	Unused



memory mapping

- **DEC** (Digital Equipment Corporation) mapped peripherals into the memory bus, so that the devices appeared to be memory locations.



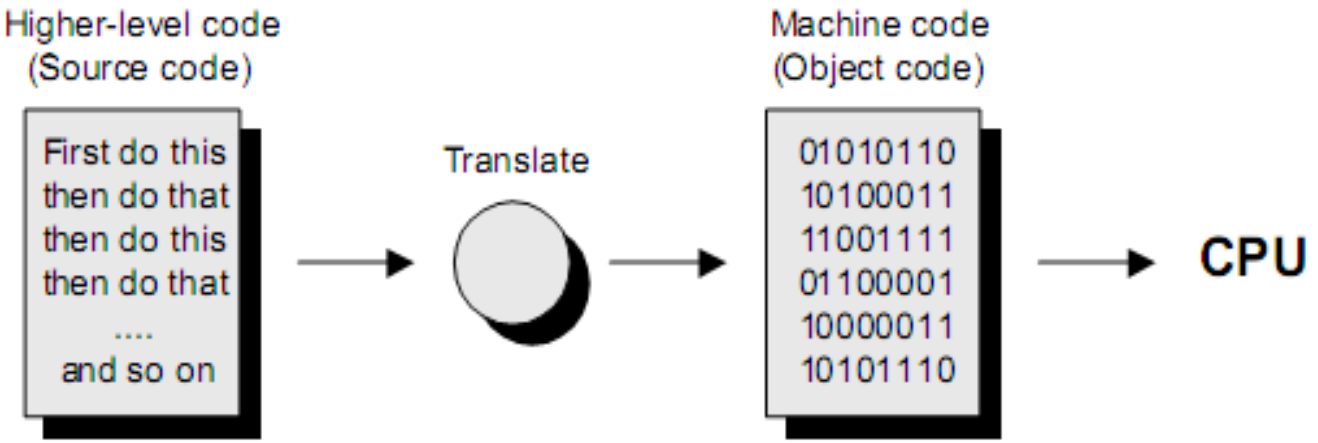
programming machine language

- feed programs and data as numerical values to the computer

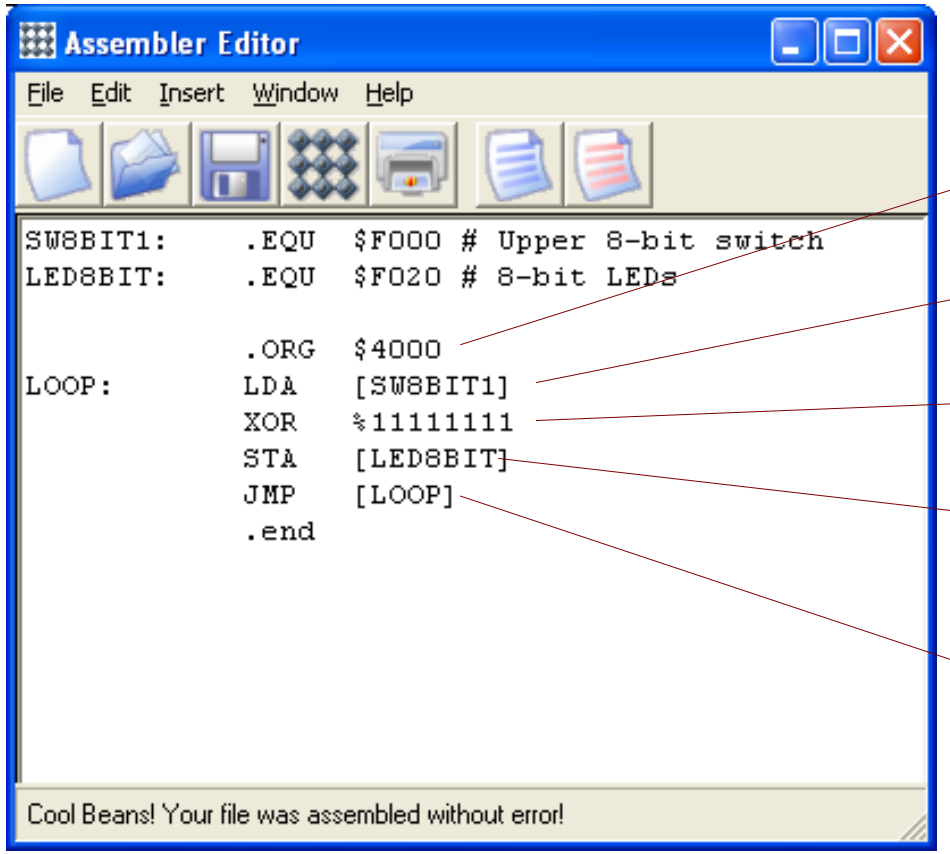
Address	Data
$4000	$91
$4001	$F0
$4002	$0
$4003	$40
$4004	$FF
$4005	$99
$4006	$F0
$4007	$20
$4008	$C1
$4009	$40
$400A	$0

programming goal

- describe at high level
- translate to machine code



programming assembly language



Address	Data
$4000	$91
$4001	$F0
$4002	$0
$4003	$40
$4004	$FF
$4005	$99
$4006	$F0
$4007	$20
$4008	$C1
$4009	$40
$400A	$0

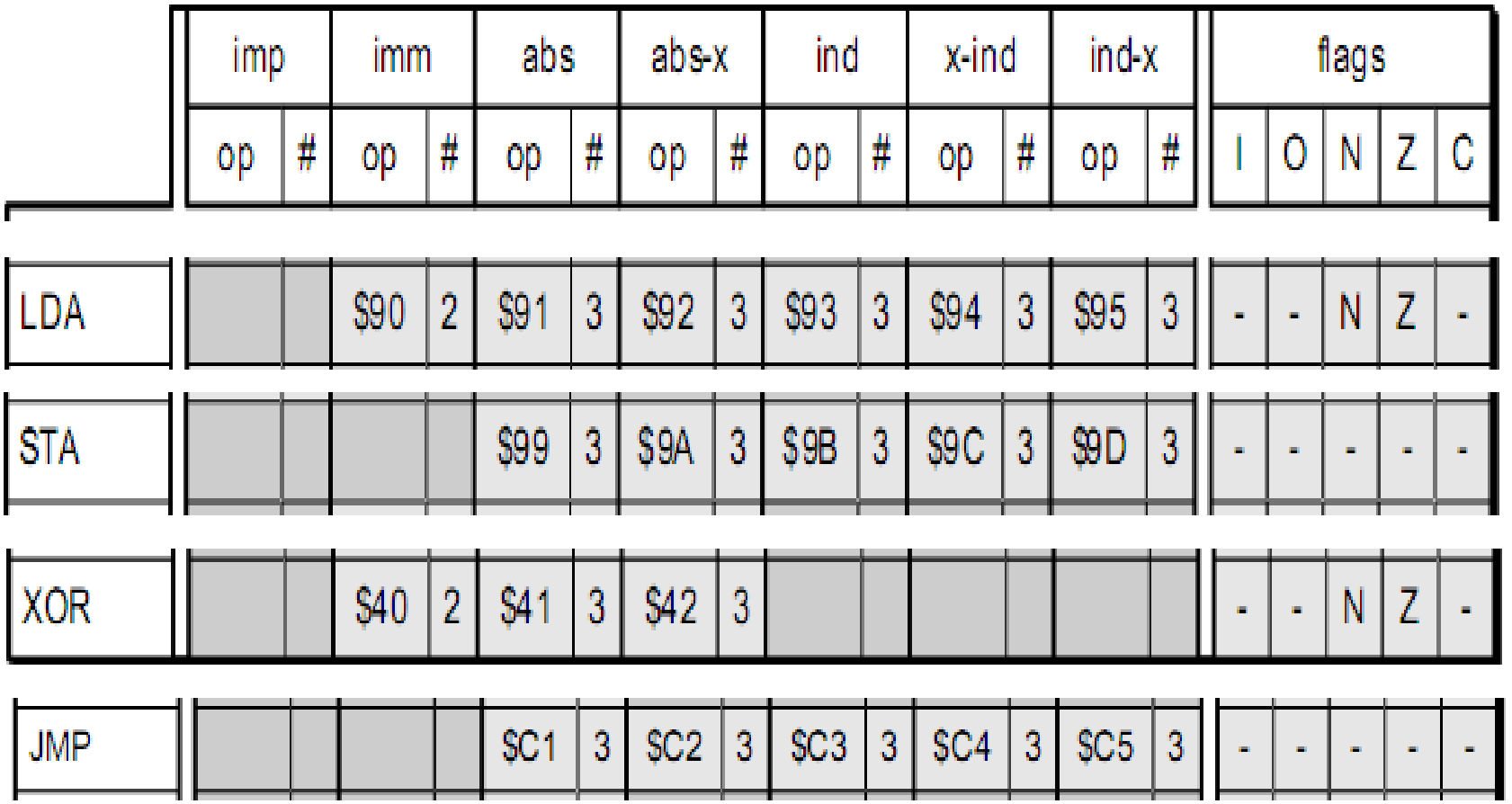
instruction set

Logical	AND	AND data in memory to the accumulator
	OR	OR data in memory to the accumulator.
	XOR	XOR data in memory to the accumulator.

Loads & Stores	LDA	Load data in memory into the accumulator
	STA	Store data in the accumulator into memory.
	BLDX	Load data in memory into the index register
	BSTX	Store data in the index register into memory.
	BLDSP	Load data in memory into the stack pointer
	BSTSP	Store data in the stack pointer into memory.
	BLDIV	Load data in memory into the interrupt vector

Jumps	JMP	Jump to a new memory location
	JSR	Jump to a subroutine.
	JZ	Jump if the result was zero.
	JNZ	Jump if the result wasn't zero.
	JN	Jump if the result was negative.
	JNN	Jump if the result wasn't negative.
	JC	Jump if the result generated a carry.
	JNC	Jump if the result didn't generate a carry.
	JO	Jump if the result generated an overflow
	JNO	Jump if the result didn't generate an overflow.

instruction set



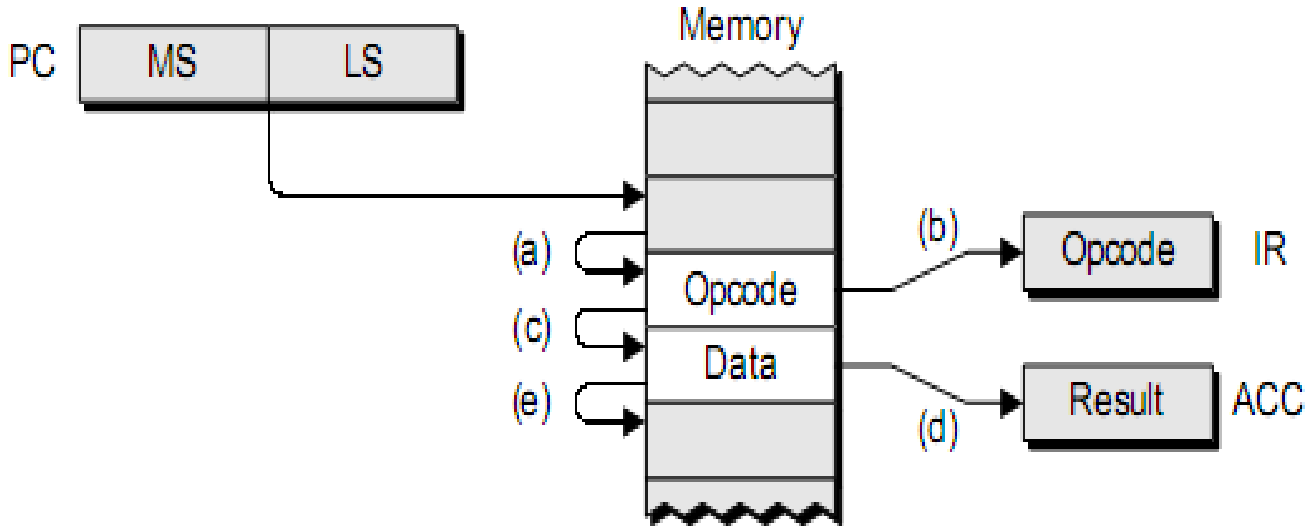
	imp		imm		abs		abs-x		ind		x-ind		ind-x		flags				
	op	#	op	#	op	#	op	#	op	#	op	#	op	#	I	O	N	Z	C
LDA			$90	2	$91	3	$92	3	$93	3	$94	3	$95	3	-	-	N	Z	-
STA					$99	3	$9A	3	$9B	3	$9C	3	$9D	3	-	-	-	-	-
XOR			$40	2	$41	3	$42	3							-	-	N	Z	-
JMP					$C1	3	$C2	3	$C3	3	$C4	3	$C5	3	-	-	-	-	-

addressing mode

- the way in which the CPU determines the address of data to be used in the instruction

addressing
immediate (imm)

- opcode is followed by
  one data operand
- e.g. XOR



addressing
absolute (abs)

- 2 address operand bytes
- fig. shows Add
- e.g. LDA, STA, JMP